

1 Základní informace

Jméno projektu	<i>Behaviour Editing Toolkit for SpringRTS</i>
Zkratka	<i>BETS</i>
Vedoucí	<i>Jakub Gemrot <gemrot@gamedev.cuni.cz></i>
Konzultanti	<i>Petr Mácha <petr.macha@bisimulations.com></i>
Anotace	<i>Cílem projektu je vytvořit rozšíření enginu SpringRTS (herní engine pro 3D real-time strategie) pro tvorbu Behaviour Trees na ovládání jednotek. Součástí projektu je vývoj uživatelsky přívětivého grafického prostředí, základních nástrojů pro ladění BT a pilotní aplikace v rámci hry NOTA.</i>

1 Motivace

Herní žánr realtime strategií (RTS) spočívá v umožnění hráči ovládat armády složené z různých druhů jednotek s různými vlastnostmi, které svádí bitvy s obdobnými armádami ovládanými soupeřem. Většina strategií však dává hráči k dispozici pouze velmi hloupé jednotky, kterým se musí doslova říkat každá akce, kterou má jednotka vykonat. Ovládání jednotek, aby dohromady plnily nějaký vyšší záměr, se běžně označuje jako tzv. „micromanagement“. Různé hry přistupují k míře složitosti akcí různě, ovšem pořád tyto akce zadává hráč. To je časově náročné a pro větší armády nepoužitelné. Proto bychom chtěli dát hráči další možnost, jak kontrolovat své jednotky skrze Behaviour Trees (BT). Chtěli bychom vytvořit nástroj, který by umožnil vytvářet a ladit složitější modely chování jednotek v uživatelsky přívětivém prostředí (chování není vhodné programovat ani v procedurálních jazycích, používají se grafické editory). Hráč by poté mohl před spuštěním zápasu definovat složitější akce a chování, podle kterých by se jednotky jeho armád měly řídit, čímž by ve výsledku dokázal efektivně řídit větší armády. I v případě, že by se takovýto nástroj neujal jakožto plnohodnotná herní mechanika, vytvářené modely chování jednotek by mohly být sdíleny v rámci herní komunity, případně by mohl být nástroj využit přímo i jako nástroj pro vývoj AI pro RTS.

2 Popis projektu

Cílem projektu je vytvořit uživatelsky přívětivý editor pro tvorbu BT řídicích chování jednotek ve hrách implementovaných v enginu SpringRTS (herní engine se zaměřením na RTS, <https://springrts.com>). To umožní komplexní reakce jednotek na hráčovy příkazy během hry. Součástí projektu bude ukázka napojení editoru na již existující hru NOTA (<http://nota.machys.net>) a vytvoření vzorových BT pro základní jednotky. Zpětná vazba práce s editorem bude získávána zpřístupňováním aktuálních verzí komunitě hráčů (komunikace s komunitou bude mediována konzultantem, Petrem Máchou, který je zároveň hlavní vývojář hry NOTA).

Ovládání

Každá jednotka bude ovládána svým skupinovým BT na základě vjemů z prostředí a hráčových příkazů. Příkaz, který je definován v BT jednotky, může být doplněn parametrem (pozice cesty, jiného objektu v herním světě) pomocí herní mapy. To bude realizováno pomocí herního widgetu, který bude zpracovávat vstupy vyznačené v mapě pomocí kurzoru myši (uživatel bude moci jednotlivé příkazy parametrizovat). Dalším prvkem ovládání je organizování jednotek do skupin. Skupině je přiřazen BT. Hráč může vydávat příkazy celým skupinám.

Behaviour Trees

Rozhodovací strom ovládá jednotky na základě příkazů hráče a sensorických vstupů. Sestává se z vlastního stromu, kontextu a senzorů.

Strom je periodicky vyhodnocován v rámci tiků. Uzly jsou vyhodnocovány v rámci stromové struktury (přičemž konkrétní pořadí může záležet na typech uzlů) a každý uzel vrací jeden z (minimálně) tří stavů: success, failure, running. S těmito stavy poté pracují jejich předci.

Strom má tři typy uzlů: logické uzly, uzly s odkazem a akční uzly. Logické uzly umožňují řídit průběh vyhodnocování stromu. Uzly s odkazem umožňují využít jiný BT jakožto podstrom. Akční uzly jsou listy, které manipulují s herním světem, vyčítají senzory, ukládají nebo čtou data z kontextu nebo vyhodnocují podmínky.

Grafický editor behaviour trees

Uživatel bude mít možnost vytvářet a editovat BT pomocí vizuálního programování, zároveň bude definovat význam všech hráčových příkazů pro jednotku, a to přímo BT stromem. K dispozici mu také slouží kontexty skupiny a týmu, které může využívat k ukládání libovolných dat a jejich využití v BT. Zároveň by měl mít možnost testovat validitu vytvořeného stromu. Pokročilí uživatelé budou moci vytvořit vlastní atomické akce a senzory pomocí Lua skriptu.

Pro účely ladění BT bude mít uživatel k dispozici sandbox testovací prostředí s nástroji pro jeho editaci (přidávání nepřátelských jednotek atd.). Navíc bude možné hru dočasně pozastavovat a přehrávat ji dopředu po menších časových úsecích a resetovat. Sandbox prostředí bude taktéž založen na herním engine SpringRTS, tak aby bylo možné testovat vytvořené BT v cílovém prostředí.

Uživatel bude moci sledovat stavy stromů a obsah kontextu při běhu sandboxu. Editor se sestává z následujících komponent:

- Editor GUI
- Behaviour Trees
 - Sensory
 - Akční uzly
 - Logické uzly (sekvence, selektor, ...)
 - Uzly s odkazem (vlození externího podstromu)
- Debugging Tools
 - Breakpointy
 - Runtime reloading BT
 - Dopředný Time step
 - Sandbox environment
 - Vizualizace BT a jeho stavu

3Platforma, technologie

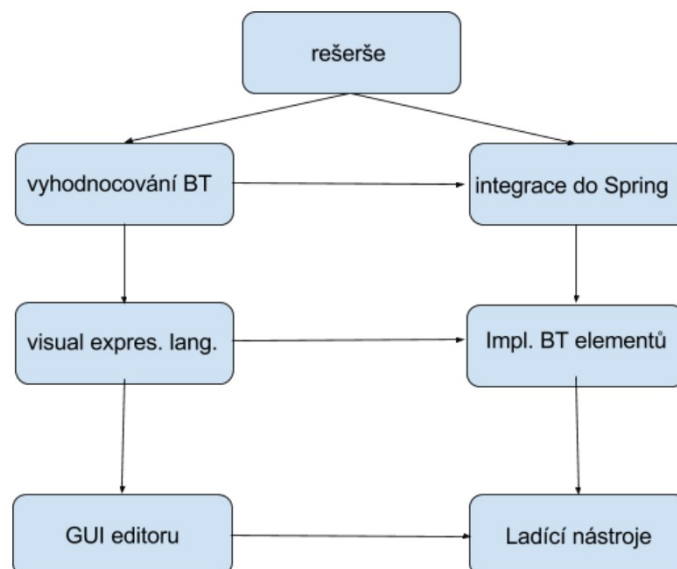
Projekt rozšiřuje framework SpringRTS. Chceme využít běžné workflow vývoje her nad daným frameworkem, a proto budeme používat kombinaci C++ a skriptovacího jazyka Lua.

Cílovou platformou budou Windows nebo Linux.

4Odhad náročnosti

Náš tým se sestává ze čtyř členů. Vývoj je rozdělen do následujících částí:

- Rešerše SpringRTS, Behaviour Trees, NOTA, návrh BT, návrh Visual expression language (10 %)
- GUI editoru (15 %)
- Integrace do SpringRTS (15 %)
- Implementace infrastruktury vyhodnocování BT (5 %)
- Implementace elementů BT (7 %)
- Implementace Visual expression language (10 %)
- Implementace ladicích nástrojů (15 %)
- Testování a příprava vzorových BT pro základní jednotky NOTA (level 1) (5 %)
- Interakce s komunitou hráčů NOTA a testování, zpřístupňování průběžných verzí a zpracovávání zpětné vazby (10 %)
- Finalizace projektu (8 %)



5Vymezení projektu

Projekt je zaměřen na následující oblasti (zaškrtněte vyhovující):

Diskrétní modely a algoritmy

	diskrétní matematika a algoritmy
	geometrie a matematické struktury v informatice
	optimalizace
Teoretická informatika	
	Teoretická informatika
Softwarové a datové inženýrství	
x	softwarové inženýrství
x	vývoj software
	webové inženýrství
	databázové systémy
	analýza a zpracování rozsáhlých dat
Softwarové systémy	
	systémové programování
	spolehlivé systémy
	výkonné systémy
Matematická lingvistika	
	počítačová a formální lingvistika
	statistické metody a strojové učení v počítačové lingvistice
Umělá inteligence	
x	inteligentní agenti
	strojové učení
	robotika
Počítačová grafika a vývoj počítačových her	
	počítačová grafika
x	vývoj počítačových her

6Poznámky

Konzultant je úspěšným řešitelem projektu NOTA Space a má mnoholeté (10+) zkušenosti s enginem SprintRTS.

Předpokládaný počet řešitelů: 4

Předpokládaný začátek projektu: srpen/září 2016

Předpokládané dokončení projektu: duben/květen 2017

Předpokládaná délka projektu: 9 měsíců